3D-MODELING OF URBAN STRUCTURES

H. Gross, U. Thoennessen, W. v. Hansen

FGAN-FOM, Research Institute for Optronics and Pattern Recognition
76275 Ettlingen, Germany
gross@fom.fgan.de

Commission III, WG III/4

KEY WORDS: LIDAR, 3D-model generation, building extraction, texturing, occlusion detection, flight trajectory, image sequences

ABSTRACT:

Three dimensional building models have become important during the past years for various applications like urban planning, enhanced navigation or visualization of touristic or historic objects. 3D-models can increase the understanding and explanation of complex urban scenarios and support decision processes. A 3D-model of the urban environment gives the possibility for simulation and rehearsal, to "fly through" the local urban structures with multiple perspective viewing, and to visualize the scene out from different viewpoints. The building models are typically acquired by (semi-) automatic processing of Laser scanner elevation data or aerial imagery. We are presenting an automatic generation method of polyhedral 3D-models from Laser height data in our paper. The methods of deriving a DTM and a DSM from the data as well as the estimation of a ground map for the built-up area as alternative of a cadastral map are especially investigated. An approach for the classification of vegetation areas is presented.
Although for some applications geometric data alone is sufficient, for visualization purposes a more realistic representation with textured surfaces is necessary. The associated textures from buildings are extracted either from airborne imagery or, especially for facades, from images taken by ground based cameras. We have investigated the selection of optimal texturing images from the acquired data including occlusions and multiple representations. Results are presented.

1. INTRODUCTION

Three-dimensional building models have become important during the past years for various applications like urban planning, enhanced navigation or visualization of touristic or historic objects [Brenner et al., 2001]. They can increase the understanding and explanation of complex scenes and support the decision process. The benefit for several applications like urban planning or the virtual sightseeing walk is demonstrated by utilization of LIDAR data.
Whereas in play games or rehearsals a virtual urban environment can be modeled, in real scenarios the models of the urban objects have to be extracted from the reality to represent the real situation. Especially in time critical situations the 3D-models must be generated as fast as possible to be available for a simulation process. That requires automatic tasks utilizing all information available in the network (e. g. images, maps, DEM, DTM). In most cases the necessary object models are not available in the simulation data base and a data acquisition has to be performed.
Different approaches to generate the necessary models of the urban scenario are discussed in the literature. Building models are typically acquired by (semi-) automatic processing of Laser scanner elevation data or aerial imagery [Baillard et al., 1999]. For large urban scenarios LIDAR data can be utilized [Thoennessen & Gross, 2002]. M. Pollefeys uses projective geometry for a 3D-reconstruction [Pollefeys, 1999] from image sequences. C. S. Fraser et al. use stereo approaches for 3D-building reconstruction [Fraser et al., 2002].
We propose a combination of the different approaches mentioned before. In LIDAR data 3D-information is directly available. Due to the vertical view of the sensor to the nadir during data acquisition, the building structures are bounded by the ground projection of the roof surfaces. We have developed algorithms for the segmentation of roof surface areas and the generation of CAD-models of gable-roofed buildings.
These common CAD-models represent the geometrical properties of the main structures of the objects. By texturing the models important additional information of an object can be provided. This could be the location of windows and doors which are of interest. The images providing the textures can be captured by a UAV or local ground based sensor systems. This requires a determination of the camera parameters to project the model surfaces onto the images. To achieve the inner and outer parameters of the camera and the track of the camera, we use the approach of projective geometry. Then image patches are cut out by projected 2D-polygons representing the faces of the 3D-model. These are used to visualize the 3D-object model with the corresponding images mapped on it as texture [Thoennessen & Gross, 2002].
One focus of the work was deriving a DTM and a DSM from the data as well as the estimation of a ground map for the built-up area as alternative of a cadastral map. Additionally the classification of vegetation areas is presented. The reconstruction of complex buildings from Laser height image data is the subject of the first two chapters. Caused by the vertical viewpoint during data recording, the building structures are bounded through the roof surfaces. Under consideration of the inclination of the roof surfaces, polyhedral models of the objects can be produced. Chapter 4 deals with the problem of texturing. In chapter 5 we present investigations to determine the trajectory of the camera and the inner parameters of the camera.

2. APPROXIMATION OF THE GROUND MAP OF BUILDINGS

If any cadastral map from the region of interest exists, the boundaries of the buildings are determined by this map. In many cases that cadastral information may be not available. Therefore the object boundaries have to be generated from the image data by subtracting the digital terrain model (DTM) from the digital elevation model (DEM).

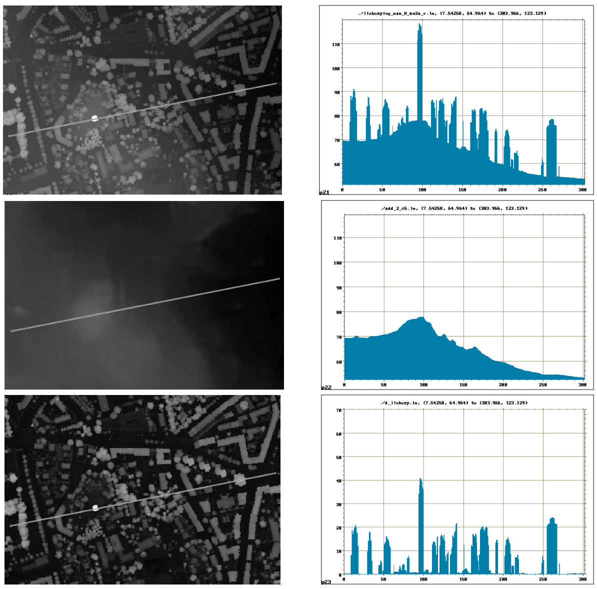
2.1 Determination of the DTM

The raw data acquired by a Laser scanner/LIDAR typically is a digital surface model (DSM), i.e. surface objects like trees or buildings are contained in the data set. This section describes how these objects can be removed in order to create a digital terrain model (DTM).
The approach is based on the observation that at object boundary an essential height jump occurs. Therefore the first step of the processing chain is the calculation of a gradient magnitude image. Then so-called essential points are marked for which the gradient magnitude exceeds a predefined threshold. The marked essential points are replaced by the minimal value in a specified neighborhood. Between two essential points the height is linearly interpolated.
This method is applied to the rows and the columns separately. To avoid artifacts and to deal with defined values (essential points) also at the image boundaries during interpolation, the first and last rows and columns respectively are processed in advance. As result we get two images: one for row and one for column-oriented processing. A convolution of the mean of both images yields the DTM. Figure 1 shows the steps from the original image to the DTM-image and the difference of both.

a

b

c



©Topeye

Figure 1. DTM determination
a) DSM and a profile along a line (horizontal and vertical scaling are different)
b) digital terrain model and profile
c) difference between DSM and DTM-image

2.2 Classification of vegetation using first-pulse/last-pulse variance

The determination of the building contour is often disturbed by vegetation - in particular if trees are occluding the roof of the building. These problems are partially solved by classification of the LIDAR data. The difference of first and last pulse signal is a significant feature for vegetation because the foliage is partially penetrable. Unfortunately the walls of buildings show a similar behavior due to the sampling mechanism of the sensor system. As a solution the shape of the classified areas is taken into account. In the case of trees the conspicuous areas of vegetation are shaped like a circle in contrast to wall boundaries which are of elongated shape.

2.3 Generation of the ground map of the buildings by recursive rectangle approximation

Many buildings are composed from parts with rectangular shape. Due to this the shape of a building can be described by a rectangular polygon.
The segmentation process for buildings delivers regions without straight boundaries caused by the variations of the data (Figure 2a). The small tower is considered by the algorithm like an own building. The boundaries of the building will be approximated by rectangular polygons to substitute the missing cadastral information.

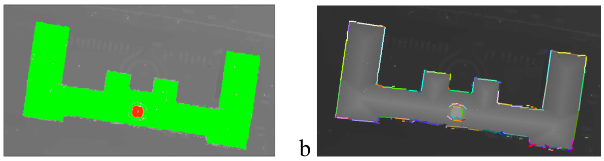


Figure 2. a) Original image with segmented object
b) Edges of the object

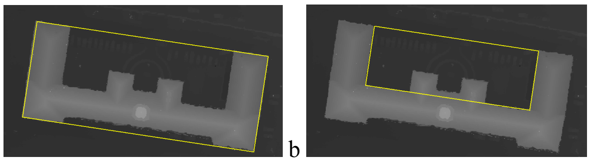


Figure 3. a) Surrounding rectangle; b) surrounding rectangle of the greatest non-building part

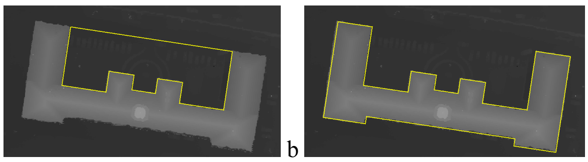


Figure 4. a) Surrounding polygon of a non-building part
b) Rectangular polygon approximation

Suppose $s := \{x \mid x = \text{point of the building}\}$ is the set of the segmented points of the building and $P_0(s)$ is the smallest surrounding rectangle (Figure 2a) with the same orientation as the orientations of the boundary edges [Burns et al., 1986] (Figure 2b). Let $A(s)$ be the size of the area.

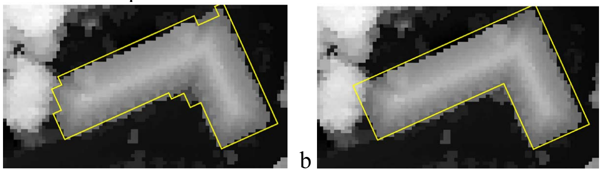
The difference $D(P_0,s) := \{y \mid y \subset P_0(s) \setminus s \wedge (y \text{ contiguous})\}$ is the set of contiguous points inside the rectangular polygon $P_0(s)$ reduced by the point set s. The cardinality of D is $N(D(P,s)) = |D(P,s)|$.

Construct $\forall y \in D(P_0,s)$ with $A(y) \geq \mathit{threshold}$ the refined rectangular polygon $P_n(s) := P_{n-1}(s) \setminus P_{N(D(P_0,y))}(y)$ with $n = 1 \ldots N(D(P_0,s))$. The same algorithm is used to determine $P_{N(D(P_0,y))}(y)$. This implies that we subtract the polygon after refining it until the required approximation quality is achieved.

Due to this method all rectangular polygons describing the areas outside the building but inside the surrounding rectangle (Figure 3b, Figure 4a) will be subtracted from the original rectangle. The determination of those outer polygons follows the same method, but by exchanging building and non-building parts alternatively. The formal recursive process does not depend on the approximation of a building or a non-building part. The result is a description of the contour of the building by a rectangular polygon (Figure 4b).

2.4 Generalization of the building ground map

If there are small convexities or indentations in the building contour, short edges are removed by modifying the object contour through generalization. The area is changed as few as possible by adding to or removing from the object rectangular subparts. The generalization repeats until all short edges are removed. Figure 5a shows the rectangular polygon after the boundary approximation, Figure 5b shows it after the generalization process.

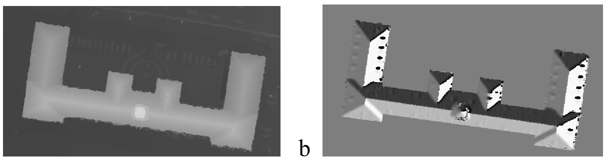


a b

Figure 5. a) Rectangular polygon
b) Generalized polygon

3. GENERATION OF THE 3D-MODEL

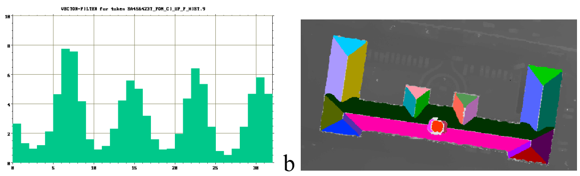
The extraction of simpler 3D-models from Laser height image data was described in [Geibel & Stilla, 2000]. The different steps of the analysis are described, for the example, in Figure 6a.



a b

Figure 6. Extraction of roof surfaces in height image data a) original image b) local orientation

Internal building pixels are those whose height difference does not exceed a predetermined threshold to the central pixel of a small subwindow.



a b

Figure 7. a) Histogram of local plane orientations
b) Segmentation result using orientation histogram

Then a detailed analysis of the roofs is enforced in these regions. The amount and orientation (Figure 6b) of the gradient is calculated by a local adaptive operator in a 3 x 3 environment. Within interrelated areas of a building an orientation histogram is produced. The histogram contributions are weighted with the value of the gradient. In Figure 7a the histogram is shown for a typical building with 4 different orientations of the roof planes. Points with the same slope contribute to the same bucket in the orientation histogram. The unification of connected points with the same slope in a specified environment defines a roof surface (Figure 7b). The roof surfaces are described by polygons afterwards. A polygon encloses the entire roof surface including disturbed areas.

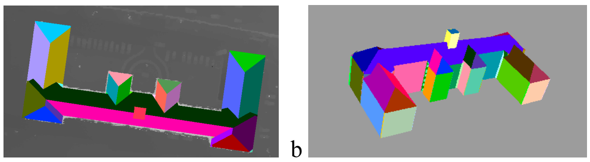
For each roof surface a plane approximation is calculated. Only points inside the circumscribing polygon are taken into account. Also holes caused by disturbances are excluded. By a least squares approach, the unknown plane parameters are determined through minimization.

These plane coefficients are determined for all roof surfaces. Disturbed values should be suppressed in order to get the best possible plane approximation. Therefore a noise threshold is determined afterwards. With the renewed calculation of the plane only those points of the roof surface, whose distance to the previously calculated plane is smaller than the mentioned threshold, are taken into account. This process is performed repeatedly until different conditions are held.

The approximated plane is the base to form a representative plane. A part of its borders is determined by the intersections of the approximated plane with its neighbor planes. The outer border is defined by the ground map of the building. In this way the roof surfaces are described in correspondence to the outer building surface. This is in accordance with a building model described by straight lines.

Polygon points near the building edges are replaced by the edge or part of it. After calculation of the intersection lines of a roof plane with its neighbors all border lines of this plane are summarized to a closed polygon. Using the plane parameters the polygon points receive also height information.

Until now only the roof surfaces of the object are described by 3D-polygons. The walls of the buildings are constructed through the outer polygon edges of the roof surfaces (upper edge) and through the terrain height (lower edge) available from the LIDAR data. Figure 8a shows the 2D-top-view of the result. Its 3D-visualization is shown in Figure 8b (re-colored for the wall representation).



a b

Figure 8. 3D-modeling of a building
a) Generated roof surfaces
b) Automatically generated building CAD-model

4. TEXTURING OF A 3D-MODEL

The complex forms of the detail-structures of urban buildings are restricted to describe objects through simple polyhedric models. A simple texturing of the models delivers important additional information on the object e.g. position of windows and doors without a detailed expensive model extension.

The texturing of 3D-building models is described as follows:

- Projection of the 3D-models onto the 2D-images,
- Dissolution of occlusion situations,
- Selection of the optimal image part for each 3D-model surface,
- Preparation of the description file for the textured 3D-model.

4.1 Projection of the 3D-model surface onto the 2D-sensor images

For the projection of a model surface onto an image, the sensor parameters position, rotation, and focal length are required. Assuming these parameters are determined automatically (see chapter 5), then on the basis of these parameters all model surfaces are transformed to the camera coordinate system.
Problems caused by model points lying behind the image plane are solved by a clipping algorithm. All points of the model are now projected in accordance with the focal length on the appropriate sensor image. Figure 9 shows the projection of the front side of the 3D-model points in accordance with the camera parameters to an image of the building from the south.

Figure 9. Projection of the generated 3D-model onto an image of the building

The front projection in Figure 9 is relatively easily processed. But this cannot be expected in any situation. Many sides of the building are not completely visible which is caused by the complex building structure and difficult conditions for data acquisition. For other object parts there exist multiple image candidates to extract a corresponding part of the texture. Furthermore problems are caused by adverse viewing angles (Figure 10).

Figure 10. Sensor images taken from the north

In order to decide which image can deliver the optimal texture for a model surface, occlusion situations are inspected additionally.

4.2 Dissolution of occlusion situations

The projection of all planes of an object onto an image for texturing encloses all planes with normal vector pointing to the camera including planes, completely or partly hidden by nearer planes. The principle drawing in Figure 11 shows that the eastern walls of middle- and west-wing are partly hidden by the east-wing. This image can only be used for texturing of the east-wing and the visible parts of the other wings.

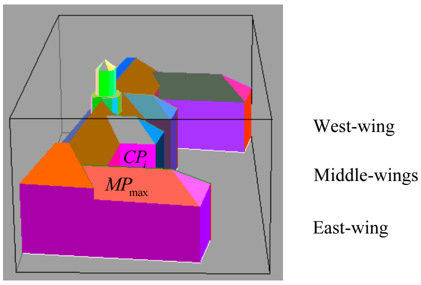


Figure 11. Occlusions of object parts by nearer objects

If the projection of the planes into the image results in an overlap area of two ore more planes, then we take the nearest one. Only for the nearest plane inside the overlap area the content of the image can be used as texture.
Let be $MP_{\max}$ the polygon with the greatest point set $CP_{\max}$ in the image. After transformation we get for all projected planes the intersection $D_i = CP_{\max} \cap CP_i \quad \forall i,\ i \neq \max$. We calculate $\forall i$ with $D_i \neq \emptyset$ the gravity points $\overline{x_{si}}$ of the intersection. By using the camera parameters the original points of the common gravity point on the 3D-building planes and their distance to the camera are determined. Comparing both distances we get the nearer plane hiding the farer ones.

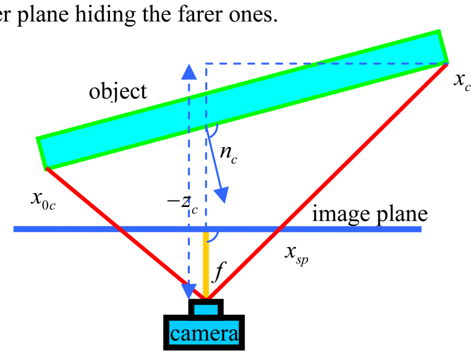


Figure 12. Coordinates of the original 3D-point

At first we project the pixel coordinates of $\overline{x_s}$ onto the camera plane. Including the focal length f this results in the 3D-point $\overline{x_{sp}} = \begin{pmatrix} x_p & z_p & -f \end{pmatrix}^T$. The following operations are done separately for each plane to be considered. Let $\overline{x_0}$ be the first 3D-point of this plane and $\overline{n}$ the normal vector in world coordinates. The transformation of the point into the camera system is $\overline{x_{0c}}$. The transformation of the normal vector is done with the same manner as that of the points, but without translation. This yields the vector $\overline{n_c}$ (cf. Figure 12). The equation $\overline{x_c} = \begin{pmatrix} x_c & y_c & z_c \end{pmatrix}^T = \overline{x_{0c}} + \lambda \overline{n_c}^{\perp} = \mu \overline{x_{sp}}$ determines the point we are looking for. Its component along the normal vector is given by the inner products $\overline{x_{0c}} \cdot \overline{n_c} = \mu \overline{x_{sp}} \cdot \overline{n_c}$. The intercept theorem postulates $\frac{-z_c}{f} = \frac{\left\|\overline{x_c}\right\|}{\left\|\overline{x_{sp}}\right\|} = \mu$. By elimination of μ using both equations we get the original coordinate components of this point onto the plane to $z_c = -f \frac{\overline{x_{0c}} \cdot \overline{n_c}}{\overline{x_{sp}} \cdot \overline{n_c}}$, $x_c = -x_p \frac{z_c}{f}$ and

$y_c = -y_p \frac{z_c}{f}$. The distance between the camera and this point is $d = \|x_c\|$.

If the distance between camera and retransformed point lying on the polygon $MP_{\max}$ is smaller than the distance of the point lying on D_i, then its point set is reduced to $CP_i := CP_i \setminus D_i$, otherwise the other point set is diminished to $CP_{\max} := CP_{\max} \setminus D_i$.
Only the remaining point set in the image is used to texture a part of the building plane. The same process is done for the smaller polygons.

4.3 Selection of the optimal image part for each 3D-model surface

In some cases, the texture image must be composed as a mosaic from different images. The selection of the images for the combination depends on an evaluation. This evaluation is influenced by the ratio of visible to total size of the projected surface, the angle, from which the camera looks at this side, which should lie near 90°, and the resolution of the object in the image.
A radiometric adaptation of the sensor images is necessary if the texture image has to be combined like a mosaic from multiple images from different sensors.

Figure 13. Textured building surfaces (north view)

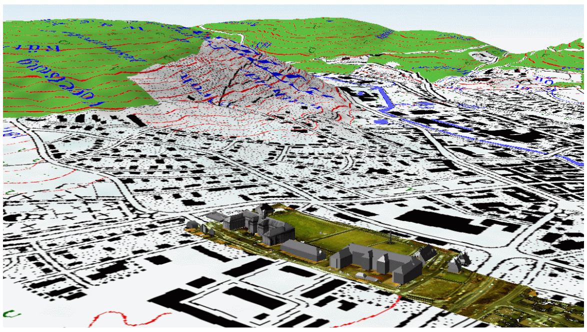


Figure 14. Textured and non-textured buildings combined with map and DTM

4.4 Preparation of the description file for the textured 3D-model

The result is written in an object description file, which is input of a 3D-visualization tool. This allows walking through the modeled built-up area virtually. Figure 13 shows the north view of the example building.
The analysis can be applied to larger scenarios with several buildings. Using a global coordinate transformation, the analysis results have been combined with a digital terrain model (DTM), maps and other information using the program system VirtualGIS of ERDAS (Figure 14). Figure 15 shows an alternative visualization combining Laser height data textured by an RGB image including analyzed buildings partly textured.

For texturing the buildings good calibration of the camera is required. In the following section a method for the determination of the parameters of the camera is explained.

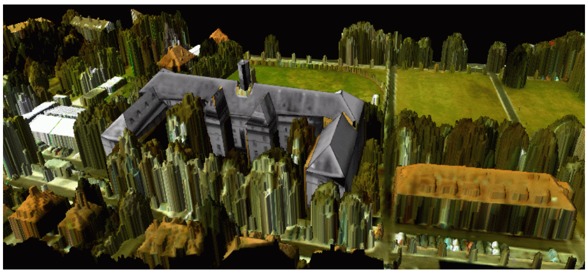


Figure 15. Textured building combined with DEM and RGB image

5. EXTRACTING FLIGHT TRAJECTORY AND 3D-MODELS FROM IMAGE SEQUENCES

In the preceding sections the creation of textured 3D models from LIDAR data and single images has been described. But the advantage of a quick and fully automatic generation of the geometric model is still hindered by the process of data fusion which is necessary to map the images correctly onto the surfaces of the 3D-models. This mapping requires the knowledge about the pose of the cameras as well as their calibration parameters. If these are not known, they have to be computed from given point assignments. Such a task – simultaneous computation of inner and outer camera parameters when no initial values are known – is commonly referred to as auto- or self-calibration [Hartley & Zisserman, 2004], [v.Hansen et al., 2004]. This task has been applied to an image sequence acquired by a UAV. In this section an approach of self-calibration and creating both model and texture from only one data source is outlined.
It is well-known among photogrammetrists and in the computer vision community that it is possible to retrieve structure from motion. Several images taken from different viewpoints or the video stream of a moving camera provide enough information to reconstruct both, the sensor pose and trajectory along with calibration parameters for the camera, and the 3D-scene viewed by the camera. In [Hartley & Zisserman, 2004] many aspects are covered in detail so that only a brief overview will be given here.
Suppose an object point is imaged by one camera so that the coordinates of its image are known. If a second camera takes an image of the same scene, what is then known about the location of that particular object point in this image? It turns out that its position is restricted to lie on a straight line – namely the image of the viewing ray of the first camera to the object point. This line is called the epipolar line and its parameters for any point are defined by the relative pose of the two cameras and their inner parameters (e.g. the focal length) which describe the image formation inside the camera. Every pair of corresponding points known thus yields one constraint. A total of at least seven corresponding points between both images are exploited to compute the fundamental matrix which expresses their mathematical relation.
To generate the full sensor trajectory for a long image sequence, the processing chain can be divided into three parts: Point tracking, initial projective reconstruction and complete reconstruction. The first part is to detect suitable image features and track their position through the sequence. The main reason is that in a typical video sequence the camera shift in space is only small from one frame to the next, but in order to retrieve

3D-information different object movements due to different depths must be visible in the images. On the other hand, since neighboring images do not change much it is easy to follow one object point through the sequence. Initial track points are generated using a point interest operator like e.g. the Foerstner operator [Foerstner 1994] (Figure 16a). Tracking of such points through the sequence is accomplished by point matching between image frames where the cross correlation coefficient of the region surrounding the points serves as similarity measure (Figure 16b). As an additional constraint for point displacements it can be exploited that two neighboring images are linked by a planar projective transform. Point tracking is the crucial part of the algorithm because any error introduced here could lead to a wrong result later on. Therefore robust schemes like RANSAC must be used for outlier detection.
Once all point tracks are completed, an initial reconstruction can be carried out. This consists of the creation of a coordinate frame for two cameras and the computation of the coordinates of some 3D-points in that frame. Two images are selected in such a way that they are sufficiently apart to form a proper stereo base, but still are connected by at least seven points so that the fundamental matrix can be computed. The two camera projection matrices can be recovered from the fundamental matrix – but not uniquely. The first camera can be chosen arbitrarily and for the second camera there are still four degrees of freedom left. Absolute location and orientation of the two cameras and their calibration cannot be determined from the images alone. The whole coordinate frame defined in this way differs from a metric coordinate frame by a projective transform. However, it is already possible to compute 3D-coordinates of the object points in the projective coordinate frame by triangulation of corresponding image points.
The two remaining tasks are the calibration of the cameras which also yields the transform from the projective to a metric reference frame and the inclusion of all other images into the model. With the introduction of constraints on the so far unconstrained inner parameters – e.g. focal length is constant for all images – it is possible to calibrate the cameras.

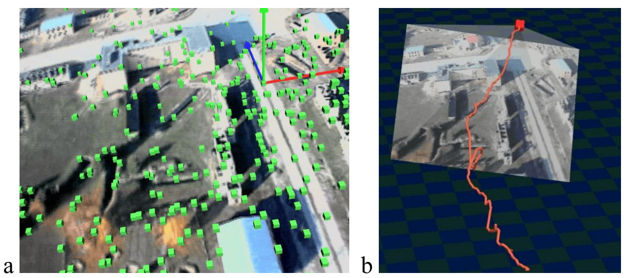


Figure 16. a) Points generated using a point interest operator b) reconstructed track of carrier

This has been done using the approach of the absolute quadric; a virtual object which is located on the plane at infinity. Its projection into the images is linked to the calibration parameters of the cameras. Using constraints, the absolute quadric can be recovered where an appropriate parameterization directly results in both, camera calibration and the transform to a metric reference frame.
Using the object points and corresponding image points already known, the camera pose can be estimated for other images through resection in space. With the additional images there are more corresponding pairs of image points so that their 3D-coordinates can be found via triangulation. Repeating these two steps it is possible to cover the complete video sequence. With known camera poses and parameters, detailed 3D-structure can be generated through a dense stereo matching. Texture information is readily available as the complete viewing geometry is known.

6. CONCLUSIONS

In LIDAR data the 3D-reconstruction of building models is directly possible. Especially in urban terrain the combined use of LIDAR data and images of other sensors is well-suited for operation planning and visualization, e.g. "fly through" visualization and detail analysis. The texturing of the different objects gives a more realistic impression and decreases modeling efforts. Especially for texturing process image sequences from UAVs can be used. From the image sequence it is possible to reconstruct both the sensor pose and trajectory.
Future work will be focused on the referencing of LIDAR DTMs and the video sequences.

LITERATURE

Baillard, C., Schmid, C., Zisserman, A. and A.Fitzgibbon, 1999. *Automatic line matching and 3d-reconstruction from multiple views.* In: ISPRS Conference on Automatic Extraction of GIS Objects from Digital Imagery, Vol. 32.

Brenner, C., Haala, N. and Fritsch, D., 2001. *Towards fully automated 3d city model generation.* In: E. Baltsavias, A. Grün and L. van Gool (eds), Proc. 3rd Int. Workshop on Automatic Extraction of Man-Made Objects from Aerial and Space Images.

Burns J. B., Hanson A. R., Riseman E. B., 1986. Extracting Straight Lines. *IEEE Transactions on Pattern analysis and Machine Intelligence* 8 (6), 425-455

Foerstner W., 1994. A Framework for Low Level Feature Extraction. *Proc. of the European conference on Computer Vision* (Vol II), Stockholm, Schweden, 383-394

Fraser C. S., Baltsavias E., Gruen A., 2002. Processing of IKONOS Imagery for Submetre 3D-Positioning and Building Extraction. *ISPRS Journal of Photogrammetry & Remote Sensing* 56, 177-194

Geibel R., Stilla U., 2000. Segmentation of Laser-altimeter data for building reconstruction: Comparison of different procedures. *International Archives of Photogrammetry and Remote Sensing*. Vol. 33, Part B3, 326-334

von Hansen W., Thoennessen U., Stilla U., 2004. Detailed Relief Modeling of Building Facades From Video Sequences. *ISPRS*, Istanbul, Turkey

Hartley R., Zisserman A., 2004. Multiple View Geometry in Computer Vision. 2nd edition, Cambridge University Press

Pollefeys M., 1999. Self-Calibration and Metric 3D-Reconstruction from Uncalibrated Image Sequences, PhD-Thesis, K. U. Leuven

Thoennessen U., Gross H., 2002. 3D-Visualization of Buildings for the Urban Warfare. *Proceedings SPIE Battlespace Digitization and Network Centric Systems III*, Orlando, Florida, USA, 1-5 Apr